

The Dissecting Power of Regular Languages

TOMOYUKI YAMAKAMI* AND YUICHI KATO*

Abstract: A study on structural properties of regular and context-free languages has promoted our basic understandings of the complex behaviors of those languages. We continue the study to examine how regular languages behave when they are “almost halving” numerous infinite languages. In particular, we are focused on a situation in which a regular language “dissects” a target infinite language into two infinite subsets. Every context-free language and its complement can be dissected by carefully chosen regular languages. By expanding the scope of our study, we show that constantly-growing languages and semi-linear languages are also dissectable; however, their complements as well as intersections are not. Under certain natural conditions, the complements and finite intersections of semi-linear languages become dissectable. Similarly, restricted to bounded languages, the intersections of finitely many bounded context-free languages and, more surprisingly, the entire Boolean hierarchy over bounded context-free languages are dissectable. As an immediate application, we show a structural property in which an appropriate bounded context-free language can separate, with infinite margins, two given infinite bounded context-free languages, one of which contains the other with an infinite margin. This property is closely related to a notion and result of Demaratzki, Shallit, and Yu (2001).

Keywords: dissecting, separating, regular language, context-free language, bounded language, semi-linear language, constant growth

1 Background Knowledge and Results’ Overview

Since the notion of context-free language was conceived and formulated as a mathematical model of natural languages by Chomsky [2, 3] in the 1950s, it has remained an intriguing research subject for almost six decades both in theory and in practice. In formal language theory, context-free languages have been of great importance in, for instance, parsing programming languages since their introduction. In an early stage of the study of context-free languages, a useful “structural” property, known as *semi-linearity*, was discovered in [10], and another useful property, dubbed later as a *pumping lemma*, was proven in [1]. The former property dictates a behavioral pattern of the times each symbol occurring inside each string of a given language, whereas the latter indicates the existence of numerous sequences of constantly-growing strings inside the language. The underlying structures of regular languages, in contrast, have been widely understood by a number of different frameworks, including the Myhill-Nerode theorem, monadic second-order logic, and finitely generated monoids.

Recently, new realms of structural properties that highlight the context-freeness of languages have been developed in an obvious connection to structural complexity issues of polynomial time-bounded complexity classes. For instance, the notions of immunity as well as pseudorandomness were introduced into context-free languages in [14]. The notion of minimal cover was also applied to regular languages in [4]. These properties have left unsolved numerous problems, concerning the structural properties of regular and context-free languages, which, we suspect, might have rooted in certain unknown natures of the languages. To promote our understanding of regular and context-free languages, it must be desirable to unearth those hidden natures. In this line of study, this paper aims at exploring another natural structural property, which we fondly name “dissectability.” This property, however, is most interesting for weak computation. One reason is that, for instance, polynomial-time computable languages are too powerful to dissect easily any “computable” language of infinite size.

Normally, regular languages are considered to be weak in recognition power; however, for certain simple tasks, they can exhibit surprisingly high power. One of such tasks is to “dissect” infinite languages in certain obvious ways. As we will give an example shortly, even computationally-hard infinite languages can be dissected into “almost halves” of infinite sizes using only the power of regular languages. More precisely, an infinite set C is said to *dissect* a target infinite set L , as illustrated in Fig.1, if two disjoint sets $C \cap L$ and $\overline{C} \cap L$ are both infinite, where \overline{C} is the complement of C . Seemingly, such dissection is one of the simplest actions to exercise when we try to analyze a basic structure of a target set. When every infinite set in a

*Affiliation: Department of Information Science, University of Fukui, 3-9-1 Bunkyo, Fukui 910-8507, Japan

language family \mathcal{C} is dissected by regular languages, we succinctly say that \mathcal{C} is *REG-dissectable*. As a quick example, let us consider a language L generated by a grammar whose productions include a special form $S \rightarrow SS$, where S is the start symbol. Although this language L could be quite hard in complexity, it can be easily dissected by a regular language composed of strings of lengths that are equal to zero modulo 3. This dissectability is explained by a fact that L contains a series of strings of lengths $2k, 2^2k, 2^3k, 2^4k, \dots$ for a certain fixed constant $k > 0$.

A typical example of REG-dissectable language is context-free languages. Through Sections 3 to 5, two wider families of languages are also discussed. Constantly-growing languages and semi-linear languages are naturally dissected by regular languages. Under certain conditions, the complements, the intersections, and the differences of semi-linear languages are also REG-dissectable using elaborate analyses of length patterns of strings inside a given language. The analyses involve a manipulation of solutions of “semi-linear” equations. Those conditions are shown to be necessary to guarantee the REG-dissectability. On the contrary, a rather obvious limitation exists for the REG-dissectability; namely, as shown in Section 3, there is a logarithmic-space computable language that cannot be dissected by any regular language. Taking a step further forward, we will show that the class of the complements of context-free languages is REG-dissectable, essentially by an application of the aforementioned pumping lemma. More surprisingly, when limited to *bounded languages* of Ginsburg and Spanier [6], we can show that the intersections of finitely many context-free languages are dissected by appropriate regular languages. This REG-dissectability result signifies the power of regular languages, because the intersections of k bounded context-free languages for $k \geq 1$ form an infinite hierarchy within the class of context-sensitive languages [9]. Our result can be obtainable, together with a result from [7], by an argument that is analogous to the argument mentioned earlier for semi-linear languages. By elaborating our argument further, we will prove that the entire *Boolean hierarchy* over the class of bounded context-free languages is also REG-dissectable. These results will be presented in Section 6. One challenging open question is to prove that the Boolean hierarchy over context-free languages is truly REG-dissectable.

The REG-dissectability notion has several connections to other notions. Earlier, Demaratzki, Shallit, and Yu [4] studied a notion of minimal cover, which means the “smallest” superset A of a given set B , where “smallest” means that there is no set between A and B with margins of infinite sizes. Motivated by their notion and results, we examine a structural property of separating two infinite nested languages with infinite margins. In our term of “separation with infinite margins” (or *i-separation*, in short), we mean, as illustrated in Fig.2, that a pair (B, A) of infinite sets, where A “covers with an infinite margin” (or *i-covers*, in short) B , can be separated by a single set C that lies in between the two sets with infinite margins. As an immediate application of the aforementioned REG-dissectability results for bounded context-free languages, we will show in Section 7 that two bounded context-free languages can be *i-separated* by bounded context-free languages in the above sense. This *i-separation* result can be further extended into any level of the Boolean hierarchy over bounded context-free languages.

2 Notions and Notations

We briefly explain a set of basic notions and notations used in the subsequent sections. We denote by \mathbb{N} the set of all *natural numbers* (i.e., nonnegative integers). For brevity, we set \mathbb{N}^+ to be $\mathbb{N} - \{0\}$. Associated with three arbitrary numbers $a, b, k \in \mathbb{N}$, we define $A_{a,b,k}$ as the set $\{an + b \mid n \in \mathbb{N}, n \geq k\}$. For any *countable* set A , the succinct notation $|A| = \infty$ (resp., $|A| < \infty$) indicates that A is an infinite (resp., a finite) set. Moreover, for two countable sets A and B , we write $A \subseteq_{ae} B$ to mean $|A - B| < \infty$, and we use the notation $A =_{ae} B$ whenever $A \subseteq_{ae} B$ and $B \subseteq_{ae} A$ hold.

We usually denote by Σ an *alphabet* (i.e., a non-empty finite set) and, for a *string* x whose symbols are chosen from Σ , we write $|x|$ to denote the *length* of x (i.e., the number of occurrences of all symbols in x).

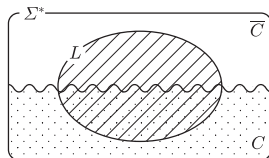


Figure 1: C dissects L .

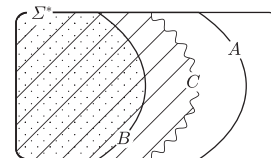


Figure 2: C *i-separates* (B, A) .

The *empty string* is always denoted λ and the length $|\lambda|$ is zero. The notation Σ^* denotes the set of all strings over Σ ; in contrast, Σ^+ expresses the set $\Sigma^* - \{\lambda\}$. A *language* over Σ is a subset of Σ^* . For a string w , w^R denotes the string w in reverse; in addition, for a language L , L^R denotes the set $\{w^R \mid w \in L\}$. The *concatenation* of two strings x and y is denoted xy . For any string x and any symbol σ , the notation $\#_\sigma(x)$ stands for the number of the occurrences of σ in x . For any language S , the *length set* of S , denoted $LT(S)$, is the collection of all lengths $|x|$ for any string x in S .

For two arbitrary languages A and B over the same alphabet Σ , the *difference* between A and B , denoted $A - B$, is the set $\{x \in \Sigma^* \mid x \in A, x \notin B\}$. The *complement* of B is the set $\Sigma^* - B$ and it is denoted \overline{B} as far as its underlying alphabet Σ is clear from the context. A language is *co-infinite* if its complement is infinite. For ease of our notations, we use the following four class operations (see, e.g., [8]): (1) $\mathcal{C} \wedge \mathcal{D} = \{C \cap D \mid C \in \mathcal{C}, D \in \mathcal{D}\}$, (2) $\mathcal{C} \vee \mathcal{D} = \{C \cup D \mid C \in \mathcal{C}, D \in \mathcal{D}\}$, (3) $\mathcal{C} - \mathcal{D} = \{C - D \mid C \in \mathcal{C}, D \in \mathcal{D}\}$, and (4) $\text{co-}\mathcal{C} = \{\overline{C} \mid C \in \mathcal{C}\}$, where \mathcal{C} and \mathcal{D} are language families.

For convenience, we write REG and CFL to denote the sets of all regular languages and of all context-free languages, respectively. The language family $\text{CFL}(k)$ (the *k-conjunctive closure* of CFL [13, 14]) is defined inductively as follows: $\text{CFL}(1) = \text{CFL}$ and $\text{CFL}(k) = \text{CFL}(k-1) \wedge \text{CFL}$ for $k \geq 2$. Liu and Weiner [9] showed that $\{\text{CFL}(k) \mid k \in \mathbb{N}^+\}$ forms an infinite hierarchy. The *Boolean hierarchy over CFL* is defined as follows: $\text{CFL}_1 = \text{CFL}$, $\text{CFL}_{2k} = \text{CFL}_{2k-1} \wedge \text{co-CFL}$, and $\text{CFL}_{2k+1} = \text{CFL}_{2k} \vee \text{CFL}$ for every $k \in \mathbb{N}^+$. Define $\text{CFL}_{\text{BH}} = \bigcup_{k \geq 1} \text{CFL}_k$. Note that $\text{CFL}_k \subseteq \text{CFL}_{k+1}$ for any index $k \in \mathbb{N}^+$. Obviously, it holds that $\text{CFL}_{2k} = \text{CFL}_{2k-1} - \text{CFL}$. Since CFL_2 coincides with $\text{CFL} \wedge \text{co-CFL}$, it holds that $\text{CFL} \cup \text{co-CFL} \subseteq \text{CFL}_2$.

To introduce a notion of (*deterministic*) *advice* that is fed to finite automata beside input strings, we adopt the “track” notation of [11]. For two symbols $\sigma \in \Sigma$ and $\tau \in \Gamma$, the notation $[\sigma]$ expresses a new symbol made up of σ and τ . On the input tape, this new symbol is written in a single tape cell, which is split into two tracks, whose upper track contains σ and the lower one contains τ . Notice that an automaton’s tape head scans two track symbols σ and τ in $[\sigma]$ at once. For two strings x and y of the same length n , $[\frac{x}{y}]$ denotes a concatenated string $[\frac{x_1}{y_1}][\frac{x_2}{y_2}] \cdots [\frac{x_n}{y_n}]$, provided that $x = x_1x_2 \cdots x_n$ and $y = y_1y_2 \cdots y_n$. An *advice function* is a function mapping \mathbb{N} to Γ^* , where Γ is an alphabet, called an *advice alphabet*. The advised language family REG/n of Tadaki et al. [11] is the collection of all languages L over certain alphabets Σ such that there exist a 1dfa M , an advice alphabet Γ , and an advice function $h : \mathbb{N} \rightarrow \Gamma^*$ for which (i) for every length $n \in \mathbb{N}$, $|h(n)| = n$ and (ii) for every string $x \in \Sigma^*$, $x \in L$ iff M accepts $[\frac{x}{h(|x|)}]$. Similarly, CFL/n was defined in [13].

Finally, we introduce a notion of “immunity.” Let \mathcal{F} be any family of languages. A language S is said to be *\mathcal{F} -immune* if S is infinite and S has no infinite subset belonging to \mathcal{F} (see, e.g., [14]).

3 How to Dissect Languages

Let us recall from Section 1 the notion of REG-dissectability. More generally, for any non-empty language family \mathcal{C} , we say that an infinite language S is *\mathcal{C} -dissectable* if there exists a language C in \mathcal{C} that dissects S (i.e., $|C \cap S| = |\overline{C} \cap S| = \infty$). A non-empty language family \mathcal{F} is said to be *\mathcal{C} -dissectable* if every infinite language in \mathcal{F} is \mathcal{C} -dissectable. Notice that this definition disregards all finite languages inside \mathcal{F} , and thus we implicitly assume that \mathcal{F} always contains infinite languages.

The choice of \mathcal{C} in the definition of \mathcal{C} -dissectability is of great importance. In particular, low-complexity languages are most interesting for dissectability. One reason is that high-complexity languages are too powerful to dissect most infinite languages. To see this fact, we will present two simple examples. In the first example, we consider the class P of all languages recognized by multi-tape Turing machines running in polynomial time. With the power of languages in P, we can dissect recursive languages of infinite size. Notationally, for a set S , we write $S(x) = 0$ (resp., $S(x) = 1$) to mean that $x \in S$ (resp., $x \notin S$).

Example 3.1 We claim that every infinite recursive language is P-dissectable. Let L be any infinite language, over an alphabet Σ , recognized by a single-tape Turing machine M that eventually halts on all inputs. For simplicity, let $\Sigma = \{0, 1\}$ and assume that $L \neq_{ae} \Sigma^*$ because, otherwise, the set $C = \{0x \mid x \in \Sigma^*\}$ easily dissects L . Now, we define C as follows. Let z_0, z_1, z_2, \dots be a standard lexicographic order of all strings. For each string x , we go through the following procedure from round 0 to round $|x|$. Initially, we set $A = R = \emptyset$. At round i , we first recover the value $C(z_i)$ by following the defining process of $C(z_i)$. We then simulate M on the input z_i within $|x|$ steps. Assume that $M(z_i) = 1$. Update A to be $A \cup \{i\}$ if $C(z_i) = 1$; let R be $R \cup \{i\}$ if $C(z_i) = 0$. Whenever either $M(z_i) = 0$ or $M(z_i)$ is not obtained within $|x|$ steps, we do nothing. After round $|x|$, if $|A| > |R|$, then define $C(x) = 0$; otherwise, define $C(x) = 1$. Clearly, C is in P.

By a diagonalization argument, we can show that $|C \cap L| = |\overline{C} \cap L| = \infty$. Therefore, every infinite recursive language can be dissected by a certain language in P.

In the second example, we will show that a simple use of *advice* makes it possible to dissect an arbitrary language even by regular languages.

Example 3.2 We claim that every language is REG/ n -dissectable. To show this claim, take any infinite language S over an alphabet Σ . Since S is infinite, the length set $LT(S)$ is also infinite. Hence, we partition $LT(S)$ into two infinite subsets, say, S_1 and S_2 ; that is, $S_1 \cap S_2 = \emptyset$, $LT(S) = S_1 \cup S_2$, and $|S_1| = |S_2| = \infty$. We also assume that $0 \notin S_1$. Now, we define an advice function $h : \mathbb{N} \rightarrow \{0, 1\}^*$ as follows: let $h(n) = 10^{n-1}$ if $n \in S_1$ and $h(n) = 0^n$ otherwise. We also define a dfa M as follows: on input $\lfloor \frac{x}{y} \rfloor$, if $y = 10^{n-1}$, then M accepts the input; otherwise, it rejects the input. Define $C = \{x \mid M \text{ accepts } \lfloor \frac{x}{h(|x|)} \rfloor\}$, which belongs to REG/ n . Obviously, for any $x \in S$ with $|x| \in S_1$, since $h(|x|) = 10^{|x|-1}$, M accepts $\lfloor \frac{x}{h(|x|)} \rfloor$. It thus holds that $|C \cap S| = \infty$. Similarly, for any $x \in S$ with $|x| \in S_2$, M rejects $\lfloor \frac{x}{h(|x|)} \rfloor$. Thus, $|\overline{C} \cap S| = \infty$ holds. In conclusion, C dissects S .

In the rest of this paper, we will focus our attention to the case of REG-dissectability. A pattern of the lengths of strings in a target language plays a key role in the REG-dissectability. We turn our attention to particular languages whose strings satisfy a certain length condition, known as a “constant growth property.” Formally, a language L is said to be *constantly growing* if there exists a constant $p > 0$ and a finite subset $K \subseteq \mathbb{N}^+$ that satisfy the following condition: for every string $x \in L$ with $|x| \geq p$, there exist a string $y \in L$ and a constant $c \in K$ for which $|x| = |y| + c$ holds. Such a language can be easily dissected by regular languages as shown below.

Lemma 3.3 *Every constantly-growing language is REG-dissectable.*

Proof. Let L be any language over an alphabet Σ . Now, we assume that L is constantly growing with a constant p and a finite set K . Let c be the maximal element in K . For each index $i \in [c]$, we define a language $L_i = \{x \in L \mid |x| \equiv i \pmod{c+1}\}$. We want to claim that there are at least two distinct indices $i_1, i_2 \in [c]$ such that $|L_{i_1}| = |L_{i_2}| = \infty$. Assume otherwise. Since $L = \bigcup_{i \in [c]} L_i$, at least one of L_i ’s is infinite. Our assumption implies that exactly one index $i \in [c]$ makes L_i infinite. We fix such an index. For each constant $j \in [c]$, define $S_{i,j} = \{y \in L \mid \exists x \in L_i [|x| = |y| + j]\}$. Since L is constantly growing, the set $S_{i,j}$ is infinite for a certain index j . This implies that $L_{i+j \bmod c+1}$ is infinite because $S_{i,j} \subseteq L_{i+j \bmod c+1}$. This contradicts the uniqueness of i . Therefore, there are at least two distinct indices $i_1, i_2 \in [c]$ such that $|L_{i_1}| = |L_{i_2}| = \infty$.

We define $C = \{x \in \Sigma^* \mid |x| \equiv i_1 \pmod{c+1}\}$. Clearly, C is regular. Moreover, $L_{i_1} \subseteq C$ and $L_{i_2} \subseteq \overline{C}$. This implies that $|C \cap L| = |\overline{C} \cap L| = \infty$. In other words, C dissects L , as required. \square

The property of constant growth is not sufficient for the REG-dissectability. For example, the language exemplified in Section 1 may not be constantly growing; however, it is REG-dissectable. For a wider application, it is therefore desirable to strengthen Lemma 3.3 slightly. In what follows, we succinctly write CGL for the family of all constantly-growing languages and use the notion of CGL-immunity.

Proposition 3.4 *Every language that is not CGL-immune is REG-dissectable.*

This proposition follows from Lemma 3.3 and the next trivial lemma. The latter lemma is also useful in proving certain closure properties in Section 4.

Lemma 3.5 *For any two infinite languages A and B , if A is REG-dissectable and $A \subseteq B$, then B is also REG-dissectable.*

Proof. This is trivial because any language that dissects A can dissect B whenever B is a superset of A . \square

By contrast, we will show an obvious limitation of the REG-dissectability. Following a convention, the notation L stands for the family of all languages that can be recognized by two-way deterministic Turing machines using a read-only input tape together with a fixed number of logarithmic space-bounded read/write work tapes. In the next proposition, we show that L contains a language that cannot be REG-dissectable. This result shows a clear limitation of the dissecting power of regular languages.

Proposition 3.6 *The language family L is not REG-dissectable.*

The proof of Proposition 3.6 requires the following technical property of unary regular languages. Recall the notation $A_{a,b,k}$ and, in addition, set $\mathcal{G} = \{(a, b, k) \mid a, b, k \in \mathbb{N}, b < a\}$ for the description of the property.

Lemma 3.7 *For any unary language S , S is regular iff there exists a finite set $G \subseteq \mathcal{G}$ for which $LT(S) = \bigcup_{(a,b,k) \in G} A_{a,b,k}$.*

Proof. Let S be any language over $\Sigma = \{0\}$.

(If-part) Let G be any finite subset of \mathcal{G} and assume that $LT(S) = \bigcup_{(a,b,k) \in G} A_{a,b,k}$. For brevity, we write $S_{a,b,k}$ for the set $\{0^n \mid n \in A_{a,b,k}\}$. It thus holds that $S_{a,b,k} = \{0^n \mid n = ai + b, i \geq k\} = \{0^{ak+b}(0^a)^i \mid i \in \mathbb{N}\}$. Clearly, $S_{a,b,k}$ is regular because a, b, k are all constants. Since G is finite and $S = \bigcup_{(a,b,k) \in G} S_{a,b,k}$, S is also regular.

(Only If-part) Since $S \in \text{REG}$, by [4, Lemma 2], there exist two integers $d \geq 0$ and $a \geq 1$ and two sets $A \subseteq \{0^i \mid 0 \leq i < d\}$ and $B \subseteq \{0^i \mid d \leq i < a + d\}$ such that $S = A + B(0^a)^*$. Note that $LT(S)$ equals the union $LT(A) \cup \{an + j \mid j \in LT(B), n \geq 0\}$. Since $A_{0,i,0} = \{i\}$ and $A_{a,j,0} = \{an + j \mid n \geq 0\}$, it suffices to define $G = \{(0, i, 0) \mid i \in LT(A)\} \cup \{(a, j, 0) \mid j \in LT(B)\}$ for the desired equality that $LT(S) = \bigcup_{(a,b,k) \in G} A_{a,b,k}$. \square

Now, we give the proof of Proposition 3.6.

Proof of Proposition 3.6. Consider the unary language $S = \{0^{n!} \mid n \in \mathbb{N}\}$ over the alphabet $\Sigma = \{0\}$. First, we want to show the following claim.

Claim 1 *S is in L*

Proof. It suffices to design a log-space Turing machine that recognizes L . On input of the form 0^m , the desired machine writes m in binary on its 1st work tape and 1 on its 2nd work tape using $O(\log m)$ cells. At each round, it reads out a number, say, n in binary on the 2nd tape and check if m is a multiple of n using the 3rd work tape as a counter up to n . If not, then the machine immediately rejects the input. Otherwise, it increases n by one (in binary) before entering the next round. If the machine does not reject until n reaches m , it accepts the input. \square

Next, we want to show that no regular language can dissect S . Assume otherwise; that is, there exists an infinite language $C \in \text{REG}$ over Σ dissects S . Lemma 3.7 guarantees the existence of a finite set G for which $LT(C) = \bigcup_{(a,b,k) \in G} A_{a,b,k}$. Without loss of generality, we can assume that $b < a$ for any $(a, b, k) \in G$.

Since $|C \cap S| = \infty$, there exists a triplet $(a, b, k) \in G$ such that $|\{m \mid \exists n \geq k[m! = an + b]\}| = \infty$. Here, we claim that $b = 0$. If $an + b = m!$ for a certain large integer $m > a$, then $m! \equiv 0 \pmod{a}$. Since $an + b \equiv b \pmod{a}$, it follows that $b \equiv 0 \pmod{a}$. Since $b < a$, b must be zero, as required. Moreover, we claim that $a > 1$. If $a = 1$, then $A_{1,b,k}$ equals $\{n + b \mid n \geq k\}$, which coincides with $\{n \mid n \geq k + b\}$. Thus, $|\overline{A}_{1,b,k}| < \infty$, and therefore $|LT(S) \cap LT(\overline{C})| < \infty$, a contradiction against $|\overline{C} \cap S| = \infty$.

Since $a > 1$ and $b = 0$, for a certain large constant k' , it holds that $\{m! \mid m \geq k'\} \subseteq A_{a,0,k}$. This implies that $|LT(S) \cap LT(\overline{C})| < \infty$, a contradiction. \square

4 Basic Closure Properties of REG-Dissectability

Before proceeding on a further exploration of the REG-dissectability of other languages, we quickly examine basic closure properties of the set of infinite REG-dissectable languages. For readability, we use the notation REG-DISSECT for the collection of all *infinite* languages that are REG-dissectable. Although this family REG-DISSECT is related to REG, it embodies clear traits that are quite different from those of REG.

We begin with a simple observation.

Lemma 4.1 *The set REG-DISSECT is closed under concatenation, reversal, Kleene star, and union.*

Proof. Let L, L_1, L_2 be any three languages in REG-DISSECT. [Union] Note that $L_1 \subseteq L_1 \cup L_2$. Since L_1 is REG-dissectable, Lemma 3.5 implies that $L_1 \cup L_2$ is also REG-dissectable. [Reversal] For L , take an infinite regular language C that dissects L . Consider the reversal C^R . Obviously, this C^R dissects L^R .

[Concatenation] Notice that the concatenation of L_1 and L_2 is $L_1L_2 = \{xy \mid x \in L_1, y \in L_2\}$. Since $L_1 \subseteq L_1L_2$, by Lemma 3.5, the REG-dissectability of L_1 leads to the REG-dissectability of L_1L_2 . [Kleene star] Note that the Kleene star $L^* = \bigcup_{i \geq 0} L^i$ contains L as a subset. Apply Lemma 3.5 to obtain the REG-dissectability of L^* . \square

Despite REG-DISSECT satisfies the closure properties listed in Lemma 4.1, it cannot be closed under intersection. More strongly, REG-DISSECT is not closed under intersection even with regular languages. This claim will be shown below.

Lemma 4.2 *REG-DISSECT is not closed under intersection with regular languages.*

Proof. Let $\Sigma = \{0, 1\}$ be our alphabet. Consider the set $D = \{0^{n!} \mid n \geq 1\}$. As we have shown earlier, D is not REG-dissectable. Now, we define two sets $A = \{0\}^*$ and $B = D \cup \{1\}^*$. It is easy to dissect A and B by regular sets $C_A = \{0^{2m} \mid m \geq 0\}$ and $C_B = \{1^{2m} \mid m \geq 0\}$. Hence, A and B are REG-dissectable. However, since $A \cap B = D$, $A \cap B$ is not REG-dissectable. Hence, REG-DISSECT is not closed under intersection with regular languages. \square

We will show two more non-closure properties of REG-DISSECT. For any alphabet Σ , a *homomorphism* f is a map from Σ to Σ^* . The domain of f can be further expanded from Σ to the whole set Σ^* by defining $f(\lambda) = \lambda$ and $f(x\sigma) = f(x)f(\sigma)$ for any $x \in \Sigma^*$ and $\sigma \in \Sigma$. Finally, set $f(L) = \bigcup_{x \in L} f(x)$. A homomorphism f is called *λ -free* if $f(\sigma) \neq \lambda$ for every $\sigma \in \Sigma$. We say that a language family \mathcal{F} is *closed under λ -free homomorphism* if, for every language $L \in \mathcal{F}$ and every λ -free homomorphism f , $f(L)$ also belongs to \mathcal{F} . Moreover, for two languages L and L' over the same alphabet, the *quotient* L/L' is the set $\{x \mid \exists y \in L' [xy \in L]\}$. We say that \mathcal{F} is *closed under quotient with regular languages* if, for every set $L \in \mathcal{F}$ and every regular language L' , the quotient L/L' is also in \mathcal{F} .

Lemma 4.3 *REG-DISSECT is not closed under λ -free homomorphism as well as quotient with regular languages.*

Proof. (1) For the non-closure property under λ -free homomorphism, we define $L = \{1^{n!} \mid n \in \mathbb{N}\} \cup \{0^{n!} \mid n \in \mathbb{N}\}$, which belongs to REG-DISSECT. Moreover, we define $h(0) = h(1) = 0$. Clearly, h is a λ -free homomorphism. The image set $h(L)$ equals $\{0^{n!} \mid n \in \mathbb{N}\}$, which can be proven to be non-REG-dissectable by an argument similar to the proof of Proposition 3.6.

(2) Next, we consider the non-closure property under quotient. We define $L = \{0^{n!}1^{n!} \mid n \in \mathbb{N}\} \cup \{1^{n!}0^{n!} \mid n \in \mathbb{N}\}$ and $L' = \{1\}^+$. Obviously, L' is regular. Note that the quotient $L/L' = \{0^{n!} \mid n \in \mathbb{N}\}$. As in (1), L/L' cannot be REG-dissectable. \square

5 Semi-Linear Languages and REG-Dissectability

Semi-linear languages are described by the behaviors of the number of occurrences of symbols in strings. This characteristic naturally makes those languages REG-dissectable. Under certain conditions, the complements as well as intersections of semi-linear languages are also dissected by regular languages. By stark contrast, without those conditions, they are no longer REG-dissectable in general.

5.1 Semi-Linear Languages

Parikh [10] discovered that the times of symbols occurring in each string in a context-free language L must satisfy some of certain linear Diophantine equations. This result inspires us to consider languages defined by those linear equations. Here, we introduce a notion of “semi-linear” languages by the following matrix formalism.

Firstly, we say that a subset A of \mathbb{N}^k is *linear* if there exist a number $m \in \mathbb{N}$ and an $(m+1) \times k$ non-negative integer matrix (called a *critical matrix*) T satisfying: for every point $v \in \mathbb{N}^k$, $v \in A$ iff an equation $(1, z_1, z_2, \dots, z_m)T = v$ holds for a certain tuple (called a *solution*) $(z_1, z_2, \dots, z_m) \in \mathbb{N}^m$. Equivalently, a linear set is a coset of a finitely generated sub-semigroup of \mathbb{N}^k for a certain $k \in \mathbb{N}$. A *semi-linear set* is a union of finitely many linear sets. Note that the set of semi-linear subsets of \mathbb{N}^k is closed under Boolean operations [6]. Secondly, we expand the notion of semi-linearity into languages. Let $\Sigma = \{\sigma_1, \sigma_2, \dots, \sigma_k\}$ be an alphabet for L . For any string x , a point $(\#_{\sigma_1}(x), \#_{\sigma_2}(x), \dots, \#_{\sigma_k}(x))$ in the space \mathbb{N}^k is denoted $\Psi(x)$.

and called *Parikh's image* of x . The *commutative image* (or *Parikh's image*) $\Psi(L)$ of L is the collection of all Parikh's images of strings in L . We say that the language L is *semi-linear* if $\Psi(L)$ is semi-linear. Notice that, since we are interested only in infinite languages, we always restrict our attention on the case of $T_j \neq O$ and implicitly assume that $T_j \neq O$ in the rest of this section. The notation SEMILIN denotes the set of all semi-linear languages.

We note that every semi-linear language L is constantly growing. This fact can be shown as follows. Actually, we need to discuss only the case where $\Psi(L)$ is linear. Take an $(m+1) \times k$ critical matrix $T = (d_{i,j})_{i,j}$ for L . For each index $i \in [m]$, let $e = \sum_{j=1}^k d_{i,j}$. For simplicity, assume that $e_i \neq 0$ for all i 's. Let n_0 denote $e_1 + 1$ and consider all strings $w \in L$ with $|w| \geq n_0$. We define K to be the set of all numbers between 1 and $\max\{e_1, e_2, \dots, e_m\}$. Let $\Psi(w) = (v_1, \dots, v_k)$ and let (z_1, z_2, \dots, z_m) be any solution for the equation $v = (1, z_1, z_2, \dots, z_m)T$. Since $|w| = e_1 + \sum_{i=1}^m e_{i+1}z_i > e_1$, not all z_i 's are zeros. Choose an index i_0 for which $z_{i_0} \neq 0$, and define $z'_{i_0} = z_{i_0} - 1$ and $z'_j = z_j$ for any other i 's. Finally, we take a string $x \in L$ satisfying that $|x| = e_0 + \sum_{i=1}^m e_{i+1}z'_i$. Clearly, there exists a constant $c \in K$ for which $|w| = |x| + c$.

Since semi-linear languages have the constant growth property, Lemma 3.3 therefore leads to the following consequence.

Lemma 5.1 *The language family SEMILIN is REG-dissectable.*

5.2 Finite Intersections of Semi-Linear Languages

Since REG-DISSECT is closed under union, Lemma 5.1 implies that, for any two languages $L_1, L_2 \in \text{SEMILIN}$, if $L_1 \cup L_2$ is infinite, then $L_1 \cup L_2$ is REG-dissectable. Next, let us consider a question of whether the intersection of finitely many semi-linear languages is REG-dissectable. Under a certain condition, it is possible to prove that this is indeed the case. For readability, we first focus on the intersection of two semi-linear languages.

Lemma 5.2 *For any two semi-linear languages L_1 and L_2 , if $L_1 \cap L_2$ is infinite and $\Psi(L_1) \cap \Psi(L_2) \subseteq \Psi(L_1 \cap L_2)$, then $L_1 \cap L_2$ is REG-dissectable.*

Proof. Let L_1 and L_2 be any two semi-linear languages over a k -letter alphabet Σ , say, $\{\sigma_1, \sigma_2, \dots, \sigma_k\}$. Assume that the intersection $L_1 \cap L_2$ is infinite and that $\Psi(L_1) \cap \Psi(L_2) \subseteq \Psi(L_1 \cap L_2)$. Hereafter, we aim at proving that $L_1 \cap L_2$ can be dissected by a certain regular language.

Consider any partition of L_1 (resp., L_2) as $L_1 = \bigcup_{i=1}^{s_1} A_i$ (resp., $L_2 = \bigcup_{i=1}^{s_2} B_i$) using languages A_1, A_2, \dots, A_{s_1} (resp., B_1, B_2, \dots, B_{s_2}) whose commutative images are linear sets. It also holds that $\Psi(L_1) \cap \Psi(L_2) = \bigcup_{1 \leq i \leq s_1} \bigcup_{1 \leq j \leq s_2} (\Psi(A_i) \cap \Psi(B_j))$. Since $\Psi(L_1) \cap \Psi(L_2)$ is infinite, there exists a pair $(j_1, j_2) \in [s_1] \times [s_2]$ that makes $\Psi(A_{j_1}) \cap \Psi(B_{j_2})$ infinite. Fix such a pair in the following argument.

By [9, Theorem 6], the intersection of finitely many linear sets can be expressed simply by an appropriate semi-linear set. Hence, there is a series of languages D_1, D_2, \dots, D_s such that $\Psi(A_{j_1}) \cap \Psi(B_{j_2}) = \bigcup_{j=1}^s \Psi(D_j)$ and all $\Psi(D_j)$'s are linear sets. Now, we choose an index $j \in [s]$ for which $|\Psi(D_j)| = \infty$, and take an $(m+1) \times k$ critical matrix $T = (d_{i,\ell})_{i,\ell}$ for D_j .

For any point $v = (v_1, \dots, v_k)$ in $\Psi(D_j)$, each element v_ℓ can be expressed as $v_\ell = d_{1,\ell} + \sum_{i=1}^m d_{i+1,\ell} z_i$ for a certain tuple $(z_1, \dots, z_m) \in \mathbb{N}^m$. We choose an index ℓ for which $\sum_{i=1}^m d_{i+1,\ell} \neq 0$. Obviously, such an index exists because, otherwise, $d_{i+1,\ell} = 0$ for all $i \in [m]$ and thus $\Psi(D_j)$ is finite, a contradiction. In what follows, we fix this index ℓ . For convenience, set $e = \sum_{i=1}^m d_{i+1,\ell}$ and write d'_i for $d_{i,\ell}$. Now, we define $C_0 = \{x \in \Sigma^* \mid \#_{\sigma_\ell}(x) - \sum_{i=1}^{m+1} d'_i \equiv 0 \pmod{2e}\}$ and $C_1 = \{x \in \Sigma^* \mid \#_{\sigma_\ell}(x) - \sum_{i=1}^{m+1} d'_i \equiv e \pmod{2e}\}$. For each string $x \in C_r$, there exists a number $u \in \mathbb{N}$ such that $\#_{\sigma_\ell}(x) - \sum_{i=1}^{m+1} d'_i = 2eu + er$, where $r \in \{0, 1\}$. This is equivalent to $\#_{\sigma_\ell}(x) = d'_1 + \sum_{i=1}^m d'_{i+1}(2u + r)$. Since $(2u, \dots, 2u)$ and $(2u+1, \dots, 2u+1)$ are legitimate choices of (z_1, \dots, z_m) for $\Psi(D_j)$, they generate two different points, say, \tilde{v}_0 and \tilde{v}_1 in $\Psi(D_j)$. Since $\Psi(D_j) \subseteq \Psi(L_1) \cap \Psi(L_2) \subseteq \Psi(L_1 \cap L_2)$ by our assumption, two corresponding strings, say, x_0 and x_1 whose Parikh's images are respectively \tilde{v}_0 and \tilde{v}_1 belong to $L_1 \cap L_2$. Note that, for each $r \in \{0, 1\}$, x_r also belongs to C_r , and thus it is in $C_r \cap (L_1 \cap L_2)$. Since u is arbitrary, it follows that $|C_r \cap (L_1 \cap L_2)| = \infty$. Since $C_0 \cap C_1 = \emptyset$, C_0 dissects $L_1 \cap L_2$. \square

The argument used in the above proof can be easily extended from the intersection of two sets $\Psi(A_i)$ and $\Psi(B_j)$ to the intersection of an arbitrary number of sets. Therefore, we finally obtain the desired result stated below.

Proposition 5.3 Let k be any number ≥ 2 . Let L_1, L_2, \dots, L_k be k semi-linear languages. If $\bigcap_{i=1}^k L_i$ is infinite and $\bigcap_{i=1}^k \Psi(L_i) \subseteq \Psi(\bigcap_{i=1}^k L_i)$, then $\bigcap_{i=1}^k L_i$ is REG-dissectable.

Without the condition $\Psi(L_1) \cap \Psi(L_2) \subseteq \Psi(L_1 \cap L_2)$ in Lemma 5.2, we cannot prove that the intersection of two semi-linear languages is REG-dissectable. More precisely, let SEMILIN(2) be the language family SEMILIN \wedge SEMILIN. To see that SEMILIN(2) is not REG-dissectable, let us consider the following example. Let $L_1 = \{0^n 1^n \mid n \in \mathbb{N}\}$ and $L_2 = \{1^n 0^n \mid n \in \mathbb{N}\} \cup \{0^{n!} 1^{n!} \mid n \in \mathbb{N}\}$. Since $\Psi(L_1) = \Psi(L_2) = \{(n, n) \mid n \in \mathbb{N}\}$, L_1 and L_2 are in SEMILIN. However, the intersection $L_1 \cap L_2 = \{0^{n!} 1^{n!} \mid n \in \mathbb{N}\}$, which belongs to SEMILIN(2), can be shown to be non-REG-dissectable by an argument similar to the proof of Proposition 3.6.

5.3 Complements and Differences of Semi-Linear Languages

Next, let us consider the complements of semi-linear languages. Unfortunately, the family co-SEMILIN is not REG-dissectable. This is easily seen as follows. Let $L = \{0^{n!} 1^{n!} \mid n \in \mathbb{N}\}$ be a language over $\Sigma = \{0, 1\}$. Since $\Psi(\bar{L}) = \mathbb{N}^2$, \bar{L} is in SEMILIN; thus, L belongs to co-SEMILIN. As noted in the previous subsection, L is not REG-dissectable.

However, under an appropriate condition, the complements of semi-linear languages are proven to be REG-dissectable.

Lemma 5.4 Let L be any co-infinite semi-linear language over an alphabet $\Sigma = \{\sigma_1, \dots, \sigma_k\}$. If $\Psi(L) \neq_{ae} \mathbb{N}^k$, then the complement of L is REG-dissectable.

Proof. Let $L \in \text{SEMILIN}$ be any co-infinite language over an alphabet $\Sigma = \{\sigma_1, \sigma_2, \dots, \sigma_k\}$. We first partition L into A_1, A_2, \dots, A_s whose commutative images are linear sets. Clearly, it holds that $\mathbb{N}^k - \Psi(L) = \bigcap_{i=1}^s (\mathbb{N}^k - \Psi(A_i))$. For each index $j \in [s]$, take an $(m+1) \times k$ critical matrix T_j for A_j and let $T_j = (d_{i,\ell}^{(j)})_{i,\ell}$. Since $\Psi(L) \neq_{ae} \mathbb{N}^k$, $\Psi(A_i) \neq_{ae} \mathbb{N}^k$ follows for each index $i \in [s]$. Note that $v \in \mathbb{N}^k - \Psi(A_j)$ iff $v \neq (1, z_1, \dots, z_m)T_j$ holds for all tuples $(z_1, \dots, z_m) \in \mathbb{N}^m$.

Here, we introduce new notations $T_j^{(\ell-)}$ and V_ℓ . For indices $\ell \in [m]$ and $j \in [s]$, the notation $T_j^{(\ell-)}$ denotes the matrix obtained from T_j by deleting the ℓ th column, and V_ℓ denotes the set of all points $v \in \mathbb{N}^{k-1}$ that satisfy the following condition: for every index $j \in [s]$ and for every tuple $(z_1, \dots, z_m) \in \mathbb{N}^m$, it holds that $v \neq (1, z_1, \dots, z_m)T_j^{(\ell-)}$. Moreover, for convenience, let V_ℓ^\perp stand for the set $\mathbb{N}^{k-1} - V_\ell$.

(1) Assume that V_ℓ is infinite for a certain index $\ell \in [k]$. Fix such an index ℓ and choose an arbitrary point $v = (v_1, \dots, v_{\ell-1}, v_{\ell+1}, \dots, v_k)$ in V_ℓ . By the definition of V_ℓ , it follows that, for any number $d \in \mathbb{N}$, a point $v^{(d)} = (v_1, \dots, v_{\ell-1}, d, v_{\ell+1}, \dots, v_k)$ induced from v always belongs to $\mathbb{N}^k - \Psi(A_j)$ for all indices $j \in [s]$. Take a string w_d whose Parikh's image is exactly $v^{(d)}$. In particular, we have $\#_{\sigma_\ell}(w_d) = d$. It holds that $w_d \in \bar{L}$, because $w_d \in L$ implies $v^{(d)} = \Psi(w_d) \in \Psi(L)$, a contradiction. Since d is arbitrary, it suffices to define a regular set C as $C = \{x \mid \#_{\sigma_\ell}(x) \equiv 0 \pmod{2}\}$. We wish to show that $|C \cap \bar{L}| = |\bar{C} \cap \bar{L}| = \infty$. When d is even, $\#_{\sigma_\ell}(w_d) = d$ implies $\#_{\sigma_\ell}(w_d) \equiv 0 \pmod{2}$; thus, $w_d \in C$. This yields the membership $w_d \in C \cap \bar{L}$. Similarly, when d is odd, we obtain $\#_{\sigma_\ell}(w_d) \equiv 1 \pmod{2}$, implying $w_d \in \bar{C}$. Hence, w_d should be in $\bar{C} \cap \bar{L}$. Since d is arbitrary, it follows that $|C \cap \bar{L}| = |\bar{C} \cap \bar{L}| = \infty$.

(2) Assume that all V_ℓ 's are finite. In particular, V_1^\perp should be infinite. Now, we fix an arbitrary point $v = (v_2, v_3, \dots, v_k)$ in V_1^\perp and consider the set B of all possible choices $(j, z_1, \dots, z_m) \in [s] \times \mathbb{N}^m$ that satisfy the equation $v = (1, z_1, \dots, z_m)T_j^{(1-)}$. Since there are only a constant number of such choices, B should be finite. Next, we define a set D of integers as $D = \{e \in \mathbb{N} \mid e = (1, z_1, \dots, z_m)T_j[1], (j, z_1, \dots, z_m) \in [s] \times \mathbb{N}^m\}$, where $T_j[1]$ denotes the first column vector of T_j . Obviously, this set D is finite. We wish to claim that, for every number $e' \in \mathbb{N} - D$, a point $v' = (e', v_2, \dots, v_k)$ falls into the set $\mathbb{N}^k - \Psi(L)$. To show this claim, we assume otherwise. There exists a particular choice (j', z'_1, \dots, z'_m) satisfying $v' = (1, z'_1, \dots, z'_m)T_{j'}$, which implies $v = (1, z'_1, \dots, z'_m)T_{j'}^{(1-)}$. In other words, (j', z'_1, \dots, z'_m) belongs to B ; thus, we conclude that $e' \in D$, a contradiction. Similar to (1), we define $C = \{x \mid \#_{\sigma_1}(x) \equiv 0 \pmod{2}\}$. It is not difficult to show that $|C \cap \bar{L}| = |\bar{C} \cap \bar{L}| = \infty$. \square

Inspired by the arguments used in the proofs of Lemmas 5.2 and 5.4, we further prove that, under a certain condition described in the following proposition, the difference between two semi-linear languages is REG-dissectable as long as the difference forms an infinite set.

Proposition 5.5 *Let L_1 and L_2 be any two infinite semi-linear languages satisfying $\Psi(L_1) \not\subseteq_{ae} \Psi(L_2)$. If $\Psi(L_1) - \Psi(L_2) \subseteq \Psi(L_1 - L_2)$ holds, then the difference $L_1 - L_2$ is REG-dissectable.*

Proof. Let L_1 and L_2 be infinite languages in SEMILIN over Σ , where $\Sigma = \{\sigma_1, \dots, \sigma_k\}$. Consider a partition A_1, A_2, \dots, A_{s_1} of L_1 so that $\Psi(L_1) = \bigcup_{i=1}^{s_1} \Psi(A_i)$ and all $\Psi(A_i)$'s are linear sets. Note that $\Psi(L_1) - \Psi(L_2) = \bigcup_{i=1}^{s_1} (\Psi(A_i) - \Psi(L_2))$. Since REG-DISSECT is closed under union by Lemma 4.1, it suffices to focus on the difference $\Psi(A_i) - \Psi(L_2)$. For notational simplicity, we henceforth assume that $L_1 = A_i$.

Take a critical matrix T for L_1 and s critical matrices S_1, S_2, \dots, S_s for L_2 , where $s \geq 1$. Let $T = (d_{i,\ell})_{i,\ell}$ and $S_j = (e_{i,\ell}^{(j)})_{i,\ell}$ for each index $j \in [s]$. Using $m_1 + m_2$ variables $z_1, \dots, z_{m_1}, w_1, \dots, w_{m_2}$ over \mathbb{N} , for each fixed index $j \in [s]$, we consider a matrix equation $(1, z_1, \dots, z_{m_1})T = (1, w_1, \dots, w_{m_2})S_j$, which is equivalent to a set of k linear Diophantine equations: $d_{1,q} + \sum_{i=1}^{m_1} d_{i+1,q} z_i = e_{1,q}^{(j)} + \sum_{i=1}^{m_2} e_{i+1,q}^{(j)} w_i$, where q ranges over the index set $[k]$. If T satisfies that, for a certain index i , $d_{i+1,q} = 0$ for all $q \in [k]$, then a point $(1, z_1, \dots, z_{m_1})T$ does not depend on the choice of z_i . To keep our proof simple, we assume that T does not satisfy this property.

Hereafter, we discuss the case where $m_2 \leq m_1$. For ease of notational complication, we assume that, in the above set of equations, w_1, w_2, \dots, w_{m_2} as well as $z_{r+1}, z_{r+2}, \dots, z_{m_1}$ ($1 \leq r \leq m_1$) are free variables and the remainders, z_1, z_2, \dots, z_r , are bound variables. In the case of $m_1 < m_2$, similarly, we set $r = m_1$. With a help of those free variables, each bound variable z_ℓ ($\ell \in [r]$) can be expressed in the form of linear polynomial, say, $p_\ell^{(j)}(w_1, \dots, w_{m_2}, z_{r+1}, \dots, z_{m_1})$ with rational coefficients.

Now, we define a set D_ℓ for each index $\ell \in [r]$ as

$$D_\ell = \{(z_1, \dots, z_{\ell-1}, z_{\ell+1}, \dots, z_r) \in \mathbb{N}^{r-1} \mid \forall j \in [s] \forall w_1, \dots, w_{m_2} \in \mathbb{N} \\ \forall z_{r+1}, \dots, z_{m_1} \in \mathbb{N} \exists i \in [r] - \{\ell\} \text{ s.t. } z_i \neq p_i^{(j)}(w_1, \dots, w_{m_2}, z_{r+1}, \dots, z_{m_1})\}.$$

In what follows, we will examine two cases separately.

(1) Assume that $D_\ell \neq \emptyset$ holds for a certain index $\ell \in [r]$. We fix such an index ℓ and choose an element $(z'_1, \dots, z'_{\ell-1}, z'_{\ell+1}, \dots, z'_r)$ from D_ℓ . For any number $d \in \mathbb{N}$, the notation $v^{(d)}$ expresses the vector $(1, z'_1, \dots, z'_{\ell-1}, d, z'_{\ell+1}, \dots, z'_r, 0, \dots, 0)T$. By the definition of D_ℓ , it follows that $v^{(d)} \neq (1, w_1, \dots, w_{m_2})S_j$ for any $j \in [s]$ and for any tuple $(w_1, \dots, w_{m_2}) \in \mathbb{N}^{m_2-1}$. Therefore, $v^{(d)}$ falls into $\Psi(L_1) - \Psi(L_2)$. By our assumption of $\Psi(L_1) - \Psi(L_2) \subseteq \Psi(L_1 - L_2)$, this implies $v^{(d)} \in \Psi(L_1 - L_2)$; thus, $L_1 - L_2$ contains a string x for which $\Psi(x) = v^{(d)}$. In particular, we choose $2u$ and $2u + 1$ as two candidates for d , where u represents a free variable, and we fix an index $q \in [k]$ satisfying $d_{\ell+1,q} \neq 0$. Note that such q exists by our choice of T .

Assume that $v^{(d)}$ is of the form (v_1, \dots, v_k) and, moreover, set $\tilde{d} = \sum_{1 \leq i \leq r, i \neq \ell} d_{i+1,q} z'_i$. When $d = 2u$, we obtain $v_d = d_{1,q} + \tilde{d} + 2ud_{\ell+1,q}$, and thus $v_d - d_{1,q} - \tilde{d} \equiv 0 \pmod{2d_{\ell+1,q}}$ holds. On the contrary, if $d = 2u + 1$, then we obtain $v_d - d_{1,q} - \tilde{d} \equiv d_{\ell+1,q} \pmod{2d_{\ell+1,q}}$. Hence, $d = 2u$ and $d = 2u + 1$ produce two different equations modulo $2d_{\ell+1,q}$. Since u is arbitrary, the set $C = \{x \in \Sigma^* \mid \#_{\sigma_q}(x) - d_{1,q} - \tilde{d} \equiv 0 \pmod{2d_{\ell+1,q}}\}$ dissects $L_1 - L_2$.

(2) Assume that $D_\ell = \emptyset$ for all $\ell \in [r]$. Choose a pair $(\ell, q) \in [r] \times [k]$ satisfying that $\sum_{1 \leq i \leq r, i \neq \ell} d_{i+1,q} \neq 0$. Such a pair actually exists because, otherwise, we obtain $d_{\ell+1,q} = 0$ for all pairs (ℓ, q) and this makes $\Psi(L_1)$ finite, a contradiction.

For simplicity, set $\tilde{d} = \sum_{1 \leq i \leq r, i \neq \ell} d_{i+1,q} \neq 0$. Our assumption implies the existence of a certain value z'_ℓ that satisfies the following condition: for every $(w_1, \dots, w_{m_2}) \in \mathbb{N}^{m_2}$ and for every $(z_1, \dots, z_{\ell-1}, z_{\ell+1}, \dots, z_{m_1}) \in \mathbb{N}^{m_1-1}$, if $z_i = p_i^{(j)}(w_1, \dots, w_{m_2}, z_{r+1}, \dots, z_{m_1})$ for all $i \neq \ell$ then $z'_\ell \neq p_\ell^{(j)}(w_1, \dots, w_{m_2}, z_{r+1}, \dots, z_{m_1})$. Depending on a number $d \in \mathbb{N}$, we use the abbreviation $v^{(d)}$ for the point $(1, \tilde{z}_1, \dots, \tilde{z}_{m_1})T$, where we set and $\tilde{z}_\ell = z'_\ell$, $\tilde{z}_i = d$ for any $i \in [r] - \{\ell\}$, and $\tilde{z}_i = 0$ for all i 's with $r + 1 \leq i \leq m_1$.

Now, we take $2u$ and $2u + 1$ as two different values of d , where u is a free variable. Assume that $v^{(d)}$ is of the form (v_1, \dots, v_k) . Similar to the argument in (1), $d = 2u$ implies $v_q = d_{1,q} + 2u\tilde{d} + d_{\ell+1,q}z'_\ell$, whereas $d = 2u + 1$ implies $v_q = d_{1,q} + 2u\tilde{d} + \tilde{d} + d_{\ell+1,q}z'_\ell$. To obtain the desired dissection, it suffices to define $C = \{x \mid \#_{\sigma_q}(x) - d_{1,q} - d_{\ell+1,q}z'_\ell \equiv 0 \pmod{2\tilde{d}}\}$. This set C dissects $L_1 - L_2$. \square

6 Context-Free Languages and Bounded Languages

Context-free languages are an important example of semi-linear languages [10]. A semi-linearity nature of context-free language will be fully exploited in certain cases of the REG-dissectability proofs later in this section. Meanwhile, we set our focal point at the REG-dissectability of $\text{CFL} \cup \text{co-CFL}$.

Proposition 6.1 *The language family $\text{CFL} \cup \text{co-CFL}$ is REG-dissectable.*

Proof. (1) Since $\text{CFL} \subseteq \text{SEMILIN}$, it immediately follows from Lemma 3.3 that CFL is REG-dissectable.

(2) Next, we wish to show that co-CFL is also REG-dissectable. Let L be any infinite language in co-CFL . Let $\Sigma = \{\sigma_1, \dots, \sigma_k\}$ be an alphabet for L . We want to show that (*) there exists an infinite subset S of L that is constantly growing. This implies that L is not CGL-immune. Proposition 3.4 thus implies the REG-dissectability of L , as required.

To show Statement (*), we need the following form of a *pumping lemma for co-CFL*, which is a direct consequence of a pumping lemma[†] for CFL, given in [1]. This lemma, however, holds only for infinite languages. For completeness, we include its proof.

Lemma 6.2 [Pumping Lemma for co-CFL] *Let L be any infinite language in co-CFL . There exists a constant p that satisfy the following: for every string $w \in L$ with $|w| \geq p$, there exist strings u, v, x, y, z such that (i) $1 \leq |vy| \leq p$, (ii) $w = uxz$, and (iii) $w' = uvxyz$ is in L .*

Proof. If \bar{L} is finite, then $L =_{ae} \Sigma^*$ and the lemma is trivially true. Hence, we assume that \bar{L} is infinite. Since \bar{L} is in CFL, we apply the pumping lemma for CFL. Take a pumping constant p and let w be any string in L with $|w| \geq p$. Consider a finite set $A_w = \{uvxyz \mid w = uxz, 1 \leq |vy| \leq p\}$ generated from w . It suffices to show that $A_w \not\subseteq \bar{L}$. Now, assume otherwise; that is, $A_w \subseteq \bar{L}$. We then apply the pumping lemma for CFL to every string r in A_w . Since $r \in A_w$, there are strings u, v, x, y, z such that $r = uvxyz$ and $r' = uxz \in \bar{L}$. Since $1 \leq |vy| \leq p$, for a certain string r , r' coincides with w . Thus, we conclude that $w \in \bar{L}$, a contradiction. Therefore, $A_w \not\subseteq \bar{L}$ follows, as required. \square

We return to the proof of Proposition 6.1. Let us choose a pumping constant p given in Lemma 6.2. This lemma produces an infinite sequence $S = \{w_1, w_2, \dots\}$ in L such that, for every index i , $|w_{i+1}| = |w_i| + c_i$ holds for a certain number $c_i \in [p]$. Clearly, S is constantly growing. This completes the proof. \square

To utilize proof techniques developed for semi-linear languages in Section 5, we focus our attention on a restricted part of context-free languages. A language L over an alphabet Σ is said to be *bounded* if there are fixed “non-empty” strings w_1, w_2, \dots, w_m in Σ^* such that L is a subset of the set $L[w_1, w_2, \dots, w_m] =_{\text{def}} \{w_1^{i_1} w_2^{i_2} \dots w_m^{i_m} \mid i_1, i_2, \dots, i_m \in \mathbb{N}\}$ [6]. Bounded languages have been frequently used in proofs of class separations: for instance, the separation between $\text{CFL}(k)$ and $\text{CFL}(k+1)$ for every $k \geq 1$ [9].

For readability, we denote by BCFL the family of all bounded context-free languages. Analogous to $\text{CFL}(k)$ and CFL_k , we can define $\text{BCFL}(k)$ and BCFL_k as well. Liu and Weiner [9] actually proved that the class $\{\text{BCFL}(k) \mid k \in \mathbb{N}^+\}$ forms an infinite hierarchy within the class of context-sensitive languages. Furthermore, we extend Parikh’s images as follows: let the *extended Parikh’s image* $\tilde{\Psi}(w)$ be $\{(i_1, i_2, \dots, i_m) \in \mathbb{N}^m \mid w = w_1^{i_1} w_2^{i_2} \dots w_m^{i_m}\}$ for any $w \in L[w_1, \dots, w_m]$. Notice that $\tilde{\Psi}(w)$ generally forms a “set” because w may have more than one expression of the form $w_1^{i_1} w_2^{i_2} \dots w_m^{i_m}$. Finally, we define $\tilde{\Psi}(L) = \bigcup_{w \in L} \tilde{\Psi}(w)$ for every bounded language L .

For bounded languages, $\tilde{\Psi}$ works as Ψ . By extending a result of [7], Ginsburg [5] presented a close relationship between bounded context-free languages L and the semi-linearity of $\tilde{\Psi}$. What we need here is a slightly weaker form of [5, Theorem 5.4.2], as stated below.

Lemma 6.3 *For any subset L of $L[w_1, \dots, w_k]$, if $L \in \text{CFL}$, then $\tilde{\Psi}(L)$ is semi-linear.*

Theorem 6.4 *For any number $k \geq 2$, $\text{BCFL}(k)$ is REG-dissectable.*

Proof. Let $L' = L[w_1, w_2, \dots, w_m]$ and let L_1, L_2, \dots, L_k be any k subsets of L' in BCFL . Assume that $L = \bigcap_{i=1}^k L_i$ is an infinite set. First, we want to claim the following.

Claim 2 $\bigcap_{i=1}^k \tilde{\Psi}(L_i) \subseteq \tilde{\Psi}(\bigcap_{i=1}^k L_i)$.

Proof. Let v be any point in $\bigcap_{i=1}^k \tilde{\Psi}(L_i)$ and fix $i \in [k]$ arbitrarily. By the definition of $\tilde{\Psi}$, there is a unique string w in L' such that $v \in \tilde{\Psi}(w)$. Since $v \in \tilde{\Psi}(L_i)$, w should belong to L_i . Since i is arbitrary, we

[†][Pumping Lemma for CFL] For any language $L \in \text{CFL}$, there exists a constant (called a pumping constant) p such that, for every $w \in L$ with $|w| \geq p$, w can be decomposed as $w = uvxyz$ such that $|vxy| \leq p$, $|vy| \geq 1$, and $w^{(i)} = uv^i xy^i z$ is in L for every $i \in \mathbb{N}$ [1].

conclude that $w \in \bigcap_{i=1}^k L_i$. It thus follows that $v \in \tilde{\Psi}(w) \subseteq \tilde{\Psi}(\bigcap_{i=1}^k L_i)$. \square

By viewing w_1, \dots, w_m as “different” symbols $\sigma_1, \dots, \sigma_m$ as done in [5], Lemma 6.3 makes it possible for us to exploit a similarity between $\Psi(w)$ and $\tilde{\Psi}(w)$. Therefore, the same type of argument developed for the proof of Lemma 5.2 can prove that L is indeed REG-dissectable. \square

Next, we discuss the REG-dissectability of the difference of two bounded context-free languages.

Proposition 6.5 *The family BCFL_2 is REG-dissectable.*

Proof. Assume that $L_1 - L_2$ is infinite. If L_2 is finite, then the proposition is trivially true. Now, we assume that L_2 is infinite. We claim the following statement.

Claim 3 $\tilde{\Psi}(L_1) - \tilde{\Psi}(L_2) \subseteq \tilde{\Psi}(L_1 - L_2)$.

Proof. Let $v \in \tilde{\Psi}(L_1) - \tilde{\Psi}(L_2)$. Note that $v \notin \tilde{\Psi}(L_2)$. Since $v \in \tilde{\Psi}(L_1)$, there exists a string $w \in L_1$ such that $v \in \tilde{\Psi}(w)$. If $w \in L_2$, then we obtain $v \in \tilde{\Psi}(w) \subseteq \tilde{\Psi}(L_2)$, a contradiction against $v \notin \tilde{\Psi}(L_2)$. Hence, $w \notin L_2$ follows. Since $w \in L_1 - L_2$, we obtain $v \in \tilde{\Psi}(w) \subseteq \tilde{\Psi}(L_1 - L_2)$, as required. \square

Similar to the proof of Theorem 6.4, the use of similarity between $\Psi(w)$ and $\tilde{\Psi}(w)$ helps apply an argument used for the proof of Proposition 5.5 to the REG-dissectability of $L_1 - L_2$. \square

Finally, we extend the above result regarding BCFL_2 to the entire Boolean hierarchy over BCFL , denoted BCFL_{BH} , where BCFL_{BH} is defined in a similar fashion to CFL_{BH} .

Theorem 6.6 *The Boolean hierarchy BCFL_{BH} is REG-dissectable.*

Our starting point of the proof of the above theorem has already proven as in Proposition 6.5 because BCFL_2 consists of the differences of two bounded context-free languages.

Proof of Theorem 6.6. Since $\text{BCFL}_{2k-1} \subseteq \text{BCFL}_{2k}$ for every $k \geq 2$, it is sufficient to prove that BCFL_{2k} is REG-dissectable for every $k \geq 1$. We show this claim by induction on k . Notice that the basis case has been shown as in Proposition 6.5. Now, let $k \geq 2$ and consider the family BCFL_{2k} . First, we show a simple fact regarding even levels of the Boolean hierarchy BCFL_{BH} .

Claim 4 *For every index $k \geq 2$, $\text{BCFL}_{2k} = \text{BCFL}_{2k-2} \vee \text{BCFL}_2$.*

Proof. Here, we want to claim that (*) for every index $k \geq 2$, $\text{BCFL}_{2k-2} \wedge \text{co-BCFL} = \text{BCFL}_{2k-2}$. Let $\mathcal{F} = \text{BCFL}_{2k-2} \wedge \text{co-BCFL}$. Since $\text{BCFL}_{2k-2} = \text{BCFL}_{2k-3} \wedge \text{co-BCFL}$ by the definition, \mathcal{F} equals $\text{BCFL}_{2k-3} \wedge (\text{co-BCFL} \wedge \text{co-BCFL})$, which is actually $\text{BCFL}_{2k-3} \wedge \text{co-}(\text{BCFL} \vee \text{BCFL})$. Since BCFL is closed under union, we have $\text{BCFL} \vee \text{BCFL} = \text{BCFL}$. Hence, it follows that $\mathcal{F} = \text{BCFL}_{2k-3} \wedge \text{co-BCFL}$; by the definition again, the right-hand side equals BCFL_{2k-2} . Therefore, Statement (*) holds.

Next, by the definition, we have $\text{BCFL}_{2k} = \text{BCFL}_{2k-1} \wedge \text{co-BCFL}$, which equals $(\text{BCFL}_{2k-2} \vee \text{BCFL}) \wedge \text{co-BCFL}$. By DeMorgan’s law, it holds that $\text{BCFL}_{2k} = (\text{BCFL}_{2k-2} \wedge \text{co-BCFL}) \vee (\text{BCFL} \wedge \text{co-BCFL})$. Using the equation (*), we obtain $\text{BCFL}_{2k} = \text{BCFL}_{2k-2} \vee \text{BCFL}_2$. \square

By the induction hypothesis, BCFL_{2k-2} is REG-dissectable. Since BCFL_2 and BCFL_{2k-2} are both REG-dissectable, Lemma 3.5 draws a conclusion that the family $\text{BCFL}_{2k-2} \vee \text{BCFL}_2$ is also REG-dissectable. By Claim 4, this family is exactly BCFL_{2k} . Therefore, BCFL_{2k} is REG-dissectable, as required for the induction. This completes the proof of Theorem 6.6 \square

7 Application: Separation with Infinite Margins

We seek an immediate application of our result regarding the REG-dissectability of languages. To describe this application, we introduce extra terminology. Given two infinite sets A and B , we say that A *covers* B with an *infinite margin* (or A is an *i-cover* of B , in short) if $B \subseteq A$ and $A \not\equiv_{ae} B$. When A i-covers B , we briefly write (B, A) and call it an *i-covering pair*. A language C is said to *separate* (B, A) with *infinite margins* (or *i-separate* (B, A) , in short) if (i) $B \subseteq C \subseteq A$, (ii) $A \not\equiv_{ae} C$, and (iii) $B \not\equiv_{ae} C$. For convenience, we use the notation $(\mathcal{D}, \mathcal{C})$ for two language families \mathcal{C} and \mathcal{D} to denote the set of all i-covering pairs (B, A)

with $A \in \mathcal{C}$ and $B \in \mathcal{D}$. We say that \mathcal{C}' *i-separates* $(\mathcal{D}, \mathcal{C})$ if, for every pair $(B, A) \in (\mathcal{D}, \mathcal{C})$, there exists a set $C' \in \mathcal{C}'$ that i-separates (B, A) .

As the starting point, by a direct construction of appropriate languages, we intend to show that CFL/n i-separates (CFL, CFL) .

Proposition 7.1 *The language family CFL/n i-separates (CFL, CFL) .*

Proof. Let (B, A) be any i-covering pair in (CFL, CFL) . Since $A - B$ is infinite, we can choose an infinite series $S = \{w_1, w_2, \dots\} \subseteq A - B$ of different lengths. Moreover, we demand that $A - B \not\subseteq_{ae} S$. Now, we define an advice function f as $f(n) = 1^n$ if $n = |w|$ for a certain string $w \in S$, $f(n) = 0^n$ otherwise. Next, we make a dfa M behave as follows: on input x of length n with advice string $f(n)$, first check if $n > 0$ and $f(n) = 1^n$; if this is indeed the case, M accepts the input; otherwise, it rejects the input. Let C be the set of all input strings that are accepted by M when the advice function f is given. Finally, we define $C' = B \cup (A \cap C)$, which belongs to CFL/n . It is not difficult to show that C' i-separates (B, A) . \square

Now, we want to apply the REG-dissection results of the previous sections to obtain several i-separation results. The following is a key lemma that bridges between REG-dissectability and i-separation.

Lemma 7.2 *Let \mathcal{C}, \mathcal{D} be any two language families. Assume that $\mathcal{C} - \mathcal{D}$ is REG-dissectable. It then holds that, for any $A \in \mathcal{C}$ and any $B \in \mathcal{D}$, if A i-covers B , then there exists a language in \mathcal{C}' that i-separates (B, A) , where $\mathcal{C}' = \{B \cup (A \cap C) \mid A \in \mathcal{C}, B \in \mathcal{D}, C \in \text{REG}\}$. Hence, \mathcal{C}' i-separates $(\mathcal{D}, \mathcal{C})$.*

Proof. Let $A \in \mathcal{C}$ and $B \in \mathcal{D}$ be two infinite languages. Let $D = A - B$. Assume that D is infinite. Our assumption guarantees the existence of a language $C \in \text{REG}$ such that C dissects D . We define $C' = B \cup (A \cap C)$. Moreover, since C dissects D , it follows that $|(A \cap C) - B| = \infty$ and $|(A \cap \overline{C}) - B| = \infty$. These imply that $B \subseteq C' \subseteq A$ and $|A - C'| = |C' - B| = \infty$. Thus, C' i-separates (B, A) . Since $C \in \text{REG}$, C' belongs to the family \mathcal{C}' . \square

Concerning bounded context-free languages, we can show the following i-separation result.

Theorem 7.3 *For any $k \geq 1$, BCFL_k i-separates $(\text{BCFL}_k, \text{BCFL}_k)$.*

Proof. We want to show that $\text{BCFL}_k - \text{BCFL}_k$ is REG-dissectable. Hence, by applying Lemma 7.2, we immediately obtain the theorem. For our purpose, we want to show that $\text{BCFL}_k - \text{BCFL}_k$ is included in BCFL_{BH} , because BCFL_{BH} is REG-dissectable by Theorem 6.4. More strongly, we want to prove that (*) for any indices $k, m \geq 1$, $\text{BCFL}_k - \text{BCFL}_m \subseteq \text{BCFL}_{\text{BH}}$.

For simplicity, let $\mathcal{F}_{k,m} = \text{BCFL}_k - \text{BCFL}_m = \text{BCFL}_k \wedge \text{co-BCFL}_m$ and $\mathcal{G}_{k,m} = \text{BCFL}_k \wedge \text{BCFL}_m$. We will show the above claim (*) by induction on $(k, m) \in \mathbb{N}^+ \times \mathbb{N}^+$. For the case $(1, 1)$, since $\mathcal{F}_{1,1} = \text{BCFL}_2$ holds by the definition, clearly $\mathcal{F}_{1,1}$ is a subset of BCFL_{BH} . Moreover, for the case $(2, 1)$, it holds that $\mathcal{F}_{2,1} \subseteq \text{BCFL}_4$ as well as $\mathcal{G}_{2,2} \subseteq \text{BCFL}_4$ because $\text{BCFL}_4 = (\text{BCFL}_2 \wedge \text{co-BCFL}_2) \vee (\text{BCFL}_2 \wedge \text{BCFL}_2) = \mathcal{F}_{2,1} \vee \mathcal{G}_{2,2}$.

For a general case (k, m) , it suffices to consider the case $(2n, 2m + 1)$. Similar to Claim 4, we can prove the next useful relation.

Claim 5 $\text{co-BCFL}_{2k+1} = \text{BCFL}_{2k-1} \vee \text{BCFL}_2$.

By Claims 4 and 5, $\mathcal{F}_{2n, 2m+1}$ equals $(\text{BCFL}_{2n-2} \vee \text{BCFL}_2) \wedge (\text{co-BCFL}_{2m-1} \vee \text{BCFL}_2)$, which can be transformed into $\mathcal{F}_{2n-2, 2m-1} \vee \mathcal{F}_{2, 2m-1} \vee \mathcal{G}_{2k-2, 2} \vee \mathcal{G}_{2, 2}$. By the induction hypothesis, there are two indices ℓ_1, ℓ_2 such that $\mathcal{F}_{2n-2, 2m-1} \subseteq \text{BCFL}_{2\ell_1}$ and $\mathcal{F}_{2, 2m-1} \subseteq \text{BCFL}_{2\ell_2}$. By applying Claim 4 repeatedly, we then obtain $\text{BCFL}_{2\ell_1} = \bigvee_{i=1}^{\ell_1} \text{BCFL}_2$ and $\text{BCFL}_{2\ell_2} = \bigvee_{i=1}^{\ell_2} \text{BCFL}_2$. Similarly, we obtain $\text{BCFL}_{2k-2} = \bigvee_{i=1}^{k-1} \text{BCFL}_2$. Hence, $\mathcal{G}_{2k-2, 2}$ equals $(\bigvee_{i=1}^{k-1} \text{BCFL}_2) \wedge \text{BCFL}_2 = \bigvee_{i=1}^{k-1} \mathcal{G}_{2, 2}$, which is included in $\bigvee_{i=1}^{k-1} \text{BCFL}_4 = \text{BCFL}_{4(k-1)}$. Thus, we obtain $\mathcal{G}_{2k-2, 2} \vee \mathcal{G}_{2, 2} \subseteq \text{BCFL}_{4k}$. It thus follows that $\mathcal{F}_{2n, 2m+1} \subseteq \text{BCFL}_{2\ell_1} \vee \text{BCFL}_{2\ell_2} \vee \text{BCFL}_{4k} = \bigvee_{i=1}^{\ell_1 + \ell_2 + 2k} \text{BCFL}_2$. As discussed before, this is equivalent to $\text{BCFL}_{2(\ell_1 + \ell_2 + 2k)}$, which is obviously included in BCFL_{BH} . Therefore, we conclude that $\mathcal{F}_{2n, 2m+1} \subseteq \text{BCFL}_{\text{BH}}$. \square

Without a restriction onto bounded languages, we prove only the following i-separation result concerning CFL.

Theorem 7.4 *CFL i-separates (CFL, REG) .*

We will give the proof of this theorem. To use Lemma 7.2, it is sufficient for us to observe the following

simple fact.

Lemma 7.5 *Assume that a language family \mathcal{C} is closed under union with regular languages. The following three statements are logically equivalent. (1) $\text{REG} - \mathcal{C}$ is REG-dissectable. (2) $\text{co-}\mathcal{C}$ is REG-dissectable. (3) $\{\Sigma^*\} - \mathcal{C}$ is REG-dissectable.*

Proof. (2) \Leftrightarrow (3). Trivial. (1) \Rightarrow (3). Trivial. (3) \Rightarrow (1). Consider $A \in \text{REG}$ and $B \in \mathcal{C}$. We define $B' = B \cup \overline{A}$. By the property of \mathcal{C} , B' belongs to \mathcal{C} . Assume that a certain $C \in \text{REG}$ dissects $\Sigma^* - B'$. This implies that $|C \cap (\Sigma^* - B')| = |\overline{C} \cap (\Sigma^* - B')| = \infty$. Hence, we have $|C \cap (A - B)| = |\overline{C} \cap (A - B)| = \infty$. We thus conclude (1). \square

Finally, we present the proof of the desired i-separation result.

Proof of Theorem 7.4. By Proposition 6.1, we obtain the REG-dissectability of co-CFL . By Lemma 7.5, this means that $\text{REG} - \text{CFL}$ is REG-dissectable. By Lemma 7.2, we can conclude that CFL i-separates (CFL, REG). \square

8 Discussions and Open Problems

We have initiated a fundamental study on the regular languages' power of dissecting given infinite languages. Although we have developed several proof techniques and proven several basic results, unfortunately, we have left unsolved a number of intriguing questions. For instance, we have shown the REG-dissectability of BCFL_k and $\text{BCFL}(k)$ for each index $k \geq 2$; however, we have not answered the following key question.

Open Problem 8.1 *Are CFL_k and $\text{CFL}(k)$ REG-dissectable for any $k \geq 2$?*

When we move our attention from CFL to two other language families, 1-C=LIN and 1-PLIN , which were introduced in [11] as natural analogues of C=P and PP , respectively, in computational complexity theory, we have no answer to the following question.

Open Problem 8.2 *Are 1-C=LIN and 1-PLIN REG-dissectable?*

Concerning the i-separation of (CFL, CFL), the following question has still awaited its answer.

Open Problem 8.3 *Does CFL i-separate (CFL, CFL)?*

Acknowledgments. The authors are grateful to Jeffrey Shallit for having drawn the authors' attention to [4] whose core concept had helped formulate an initial notion of "dissectability."

References

- [1] Y. Bar-Hillel, M. Perles, and E. Shamir. On formal properties of simple phrase-structure grammars. *Z. Phonetik Sprachwiss. Kommunikationsforsch.*, 14, 143–172, 1961.
- [2] N. Chomsky. Three models for the description of language. *IEEE Transactions on Information Theory*, 2(3), 113–124, 1956.
- [3] N. Chomsky. On certain formal properties of grammars. *Information and Control*, 2, 137–167, 1959.
- [4] M. Domaratzki, J. Shallit, and S. Yu. Minimal covers of formal languages. In *Proc. of the 15th International Conference on Development of Language Theory (DLT 2001)*, Lecture Notes in Computer Science, Springer, Vol.2295, pp.319–329, 2002.
- [5] S. Ginsburg. *The Mathematical Theory of Context-Free Languages*. McGraw-Hill, New York, 1966.
- [6] S. Ginsburg and E. H. Spanier. Bounded ALGOL-like languages. *Trans. Amer. Math. Soc.*, 113, 333–368, 1964.
- [7] S. Ginsburg and E. H. Spanier. Semigroups, Presburger formulas and languages. *Pacific J. Math.*, 16, 285–296, 1966.
- [8] L. A. Hemaspaandra and J. Rothe. Unambiguous computation: Boolean hierarchies and sparse Turing-complete sets. *SIAM J. Comput.*, 26 (1997) 634–653.

- [9] L. Y. Liu and P. Weiner. An infinite hierarchy of intersections of context-free languages. *Math. Systems Theory*, 7, 185–192, 1973.
- [10] R. J. Parikh. On context-free languages. *J. ACM*, 13, 570–581, 1961.
- [11] K. Tadaki, T. Yamakami, and J. C. H. Lin. Theory of one tape linear time Turing machines. *Theor. Comput. Sci.*, 411, 22–43, 2010. An extended abstract appeared in the Proc. of the 30th SOFSEM Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2004), Lecture Notes in Computer Science, Springer, Vol.2932, pp.335–348, 2004.
- [12] T. Yamakami. Swapping lemmas for regular and context-free languages. Available at arXiv:0808.4122, 2008.
- [13] T. Yamakami. The roles of advice to one-tape linear-time Turing machines and finite automata. *Int. J. Found. Comput. Sci.*, 21, 941–962, 2010. An early version appeared in the Proc. of the 20th International Symposium on Algorithms and Computation (ISAAC 2009), Lecture Notes in Computer Science, Springer, Vol.5878, pp.933–942, 2009.
- [14] T. Yamakami. Immunity and pseudorandomness of context-free languages. *Theor. Comput. Sci.*, 412, 6432–6450, 2011.